Unsupervised Joint PoS Tagging and Stemming for Agglutinative Languages

NECVA BÖLÜCÜ and BURCU CAN, Department of Computer Engineering

The number of possible word forms is theoretically infinite in agglutinative languages. This brings the out-of-vocabulary (OOV) issue for part-of-speech (PoS) tagging in agglutinative languages. Since the inflectional morphology does not change the PoS tag of a word, we propose to learn stems along with PoS tags simultaneously. Therefore, we aim to overcome the sparsity problem by reducing the word forms into their stems. We adopt a Bayesian model that is fully unsupervised. We build a Hidden Markov Model for PoS tagging where the stems are emitted through hidden states. Several versions of the model are introduced in order to observe the effects of the different dependencies throughout the corpus; such as the dependency between stems and PoS tags or the dependency between PoS tags and affixes. Additionally, we use neural word embeddings to estimate the semantic similarity between the word form and the stem. We use the semantic similarity as prior information to discover the actual stem of a word since the inflection does not change the meaning of a word. We compare our models with other unsupervised stemming and PoS tagging models on Turkish, Hungarian, Finnish, Basque, and English. The results show that a joint model for PoS tagging and stemming improves upon an independent PoS tagger and stemmer in agglutinative languages.

Additional Key Words and Phrases: Unsupervised learning, part-of-speech (PoS) tagging, stemming, joint learning, neural word embeddings, Hidden Markov Models (HMM)

ACM Reference format:
Necva Bölücü and Burcu Can. 2016. Unsupervised Joint PoS Tagging and Stemming for Agglutinative Languages. 1, 1, Article 1 (January 2016), 22 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Part-of-speech (PoS) tagging is the task of assigning each word a part-of-speech tag such as noun, verb, adjective, or adverb in a given sentence. It is one of the fundamental tasks in Natural Language Processing (NLP). Many applications in NLP such as sentiment analysis, question answering, text summarization, and machine translation require PoS tagging for all languages. For example, in order to translate *saw* into another language, the translation of the word will be determined based on its PoS tag (i.e. *saw* is a tool if it is a noun and *saw* is the action *to see* if it is a verb).

Most of the work on PoS tagging [15, 19, 48] is word-based and does not expect the morphological segmentation of words. In this article, we aim to use the stems of the words for PoS tagging in order to overcome the out-of-vocabulary (OOV) issue.

Stemming is the task of reducing a word to its stem by stripping off the inflectional suffixes attached, if exists. For example, *bookkeepers* is reduced to *bookkeeper* when the word is stemmed. However, *-er* still remains because it is a derivational suffix. Various methods have been applied for stemming. Current stemming methods [10, 27, 36, 37]

usually do not expect any syntactic information. However, the PoS tag of the word is the same as the PoS tag of its stem and this information can be utilized in stemming. For example, the word *koyun* is stemmed as *koy-(mak)* (*to put*) if the word is a verb and stemmed as *koyun* (*sheep*) if the word is a noun.

PoS tagging has many drawbacks. One of the drawbacks is the ambiguity. Words may belong to different parts-of-speech depending on their syntactic roles in a given sentence. The correct PoS tag of a word also helps to find the stem of the word. For example,

- Aydınlık gelecek günler bizi bekliyor. (Bright days in the future are waiting for us.)
- Ahmet birazdan gelecek. (Ahmet will be back soon.)

gelecek in the first sentence is an adjective and the stem is *gelecek* (*future*). In the second sentence, *gelecek* is a verb and the stem is *gel-(mek)* (*to come*). This shows that PoS tagging plays an important role in stemming.

It is clearly seen that stemming and PoS tagging are two related tasks. These tasks have been usually performed as a pipeline process in various NLP applications. Stemming is usually followed by PoS tagging. One drawback of pipeline approaches is the error propagation through the successive stages in the pipeline process. Joint models can avoid this kind of problem and also leads to a better performance since the two tasks provide information for each other [40, 44, 45]. In this article, we propose to perform PoS tagging and stemming jointly in an unsupervised framework.

Joint learning also helps to reduce the sparsity in the corpus. For example, let the word *kitapçıdayken* occur only once in the corpus. It is hard to guess its tag by using only a single contextual information. However, in a joint task that involves stem-based PoS tagging, we have a higher chance to come across the same stem *kitapçı* in different inflected forms. The sparsity is even more severe in agglutinative languages. In those languages, morphemes can denote the number, gender, person, tense, and so on [25]. Turkish, Finnish, and Hungarian are examples to agglutinative languages. For example, word *gittiler* (*they went*) is split as *git+ti+ler*. Here, *git* (*to go*) is the root, *-ti* denotes the past tense, and *-ler* denotes the third person plural. Therefore, tagging the complete word forms brings the sparsity issue in agglutinative languages.

Joint PoS tagging and stemming helps to tackle the sparsity issue by reducing the dictionary size in agglutinative languages. For example;

- Yürüdüğümüz/Adj **yol/Noun** bitmiş/Verb ,/Punc daha/Adv dar/Adj bir/Det sokak/Noun açılmıştı/Verb önümüzde/Noun ./Punc *(The road we walked through was over and there was a narrower road ahead.)*
- Ama/Conj **yolu/Noun** bilmiyorum/Verb ./Punc *(But I do not know the way.)*
- Dar/Adj **yollarda/Noun** koşarak/Adv giden/Adj Kerem'i/Noun yakaladım/Verb ./Punc *(I caught Kerem running in narrow roads.)*

yolu (*the road* - accusative case) and *yollarda* (*on the roads*) are inflected forms of the same stem *yol* (*the road*). These words are all tagged with the same PoS tag in the given sentences. Using the stem *yol* rather than its inflected forms will solve the sparsity problem in tagging.

In this article we introduce a joint model for PoS tagging and stemming that extends the word-based Bayesian HMM model by Goldwater and Griffiths [19]. We introduce several different versions of the model that incorporates different dependencies, such as the dependency between the stem and the PoS tag and the dependency between the PoS tag and the affix. Additionally, we incorporate the neural word embeddings to learn the stem of a word since the inflection does not change the meaning.

Our approach mainly addresses agglutinative languages in a fully unsupervised framework. However, we did experiments also on non-agglutinative languages such as English and Basque, in addition to three agglutinative

languages: Turkish, Hungarian and Finnish. To our knowledge, this study is the first attempt in joint learning of PoS tagging and stemming in a fully unsupervised framework.

This article is organized as follows: In Section 2, the related work on PoS tagging and stemming is addressed, Section 3 describes the baseline PoS tagging model and the novel joint models for PoS tagging and stemming, Section 4 presents the experimental results and Section 5 provides a discussion on the results obtained from different languages. Finally Section 6 concludes the article with general findings of the study and the potential future work.

2 RELATED WORK

In this section, first we present the related work on PoS tagging and stemming that handle the two tasks independently as two separate tasks. Then we review the joint models on PoS tagging that combine the tagging task with other tasks such as morphological segmentation. We mainly focus on unsupervised models in the literature, since the focus of this article is solely on unsupervised learning.

2.1 Related Work on PoS Tagging

PoS tagging has been widely seen as a clustering problem. Brown et al. [9] introduce a class-based n-gram model that uses a greedy hierarchical clustering algorithm to learn the syntactic classes of the words. The contextual information is incorporated in terms of n-grams. Initially, each word is assigned to a single class. Then, each cluster pair that yields the minimum loss in the average is merged until all clusters are merged under a single cluster. Finally, a binary tree is built, which represents the hierarchy between the syntactic categories.

Schütze [42] uses Singular Value Decomposition (SVD) to reduce the dimensionality of the context matrix, which is constructed by the word vectors obtained from the two left and two right neighbour words of each word. Then, Buckshot clustering [13] is applied to cluster the words using the contextual information.

Biemann [8] applies a graph clustering algorithm, Chinese Whispers, using the 4-word context windows and the top frequent words as features.

Some other approaches see PoS tagging as a sequence labelling problem. Most of the studies in this class adopt Hidden Markov Models (HMMs) for the labelling problem.

Merialdo [32] introduces a triclass Markov model. In the study, different parameter estimation methods are compared for different sizes of training data. Relative frequency training is used for the tagged data and Maximum Likelihood training is used for the data without tags.

Banko and Moore [6] introduce the contextualized HMM tagger that emits each word from three adjacent tags including the previous and following words' tags, and not only from the current word's tag. This model involves more contextual information compared to the basic HMM.

Johnson [23] compares different parameter estimators used in HMM-based PoS tagging. For that purpose, Expectation Maximization (EM) [7], Variational Bayes [**?**] and Gibbs sampling [16] are compared. The study reveals the low-performance of the EM algorithm compared to Gibbs sampling and the Variational Bayes estimator.

Goldwater and Griffiths [19] describe a Bayesian PoS tagger that adopts a HMM with symmetric Dirichlet priors over transition and emission distributions that are distributed with Multinomial distribution. Gibbs sampling is used for the inference. Two experiments are presented in the study: one using a dictionary that contains the possible PoS tags for each word in a semi-supervised framework and one in a fully unsupervised learning framework.

All previous work assumes that the number of PoS tags are known a priori. Van Gael et al. [48] introduce infinite HMM (iHMM), which learns the number of hidden states (the number of PoS tags in the model). The model is non-parametric Bayesian and uses Pitman-Yor process to learn the transitions and emissions with Dirichlet priors. Beam sampling [51] is applied for the inference. The model is evaluated on shallow parsing as for the extrinsic evaluation.

Christodoulopoulos et al. [12] compare seven different PoS tagging models and show that older clustering-based models also perform surprisingly well compared to the recent models that see tagging as a labeling problem.

Stratos et al. [46] assume that each tag is associated with at least one word that cannot have any other tag. For example, *the* can be tagged only as a determiner and cannot have any other tag. This specific type of HMM is called an anchor HMM in their study. Non-negative matrix factorization framework [3] is extended for the parameter estimation in their model.

2.2 Related Work on Stemming

The current stemming algorithms are usually categorized in three classes: rule-based, hybrid, and statistical stemming algorithms. Rule-based stemmers learn the stems by using manually defined rules. Some of the well-known rule-based stemmers are Lovins [26], Porter [39], and Krovetz [25]. Rule-based stemming algorithms are usually supervised since they require manual definition of the rules.

Hybrid stemming algorithms combine rule-based and statistical methods in a single framework. Some of the hybrid stemming algorithms are by Shrivastava et al. [43], Goweder et al. [20], and Adam et al. [1].

Statistical stemming algorithms use statistical methods to learn the stems. Xu and Croft [52] present a method that uses the word co-occurrence statistics to cope with the drawbacks of the Porter stemmer [39]. Based on the co-occurence statistics, they implement a graph-partitioning algorithm to reduce the number of the classes that are generated by the Porter stemmer [39].

Goldsmith [17, 18] proposes an unsupervised stemming model, *Linguistica*, which is based on the Minimum Description Length (MDL) principal. The model is designed especially for morphological segmentation. However, it is used as a stemmer as well. The segmentation points in each word are found in a way to minimize the total compressed length of the corpus.

A graph-based algorithm for stemming is proposed by Bacchin et al. [4]. The algorithm splits each word at all possible split points in the first step to have a set of substrings. In the second step, a directed graph is built by using the set of substrings. Finally, the graph is used to compute the prefix and suffix scores based on the frequency of substrings.

Melucci and Orio [31] present an HMM based stemmer. States correspond to prefixes and suffixes. Transitions correspond to the rules. Expectation Maximization (EM) algorithm is used to estimate the parameters. Once the parameters are estimated, segmentation is done according to the path having the maximum probability.

McNamee and Mayfield [29] present an alternative stemming algorithm that is based on n-grams. Bigrams and trigrams are generated for each word. The approach posits that similar words share a high proportion of n-grams.

Bacchin et al. [5] extend the graph-based stemmer introduced in Bacchin et al. [4]. The extended model discovers the stems and derivations using the mutual reinforcement relationship between the stems and the suffixes. Initially, a set of possible substrings are generated by splitting each word at all positions. Then, a directed graph is built, where nodes represent the substrings. A directed edge is inserted between node x and node y if there is a word z such that $z = xy$. The estimation of affix scores is calculated by the HITS algorithm [24]. Once the prefix and suffix scores are

estimated, the algorithm finds the most probable split point by maximizing the likelihood of the prefix and suffix pairs that belong to the words in the word list.

Another stemming algorithm called YASS (Yet Another Suffix Striper) is presented by Majumder et al. [27]. It is based on a clustering algorithm that uses a string distance measure. The string distance measure is used to estimate the morphological similarity between words.

Peng et al. [37] describe a stemmer that uses the distributional similarity between words. This stemmer is applied for the information retrieval task to improve the retrieval scores.

GRAph-based Stemmer (GRAS) is introduced by Paik et al. [36]. It is a statistical stemmer that uses lexical information to group the words. Words are represented by the nodes of a graph. Weighted edges represent the frequency of the suffix pair between the vertices. The algorithm finds the related words by decomposing the graph.

Brychcín and Konopík [10] present the High Precision Stemmer (HPS), which utilizes the orthographic and semantic information as features to split the words into their stems and suffixes. The method works in two steps: First, orthographically and semantically similar words are clustered by using the Maximum Mutual Information (MMI). Second, maximum entropy classifier is performed on the clusters obtained from the first step. HPS is also used in information retrieval to improve the retrieval scores.

2.3 Related Work on Joint Models for PoS Tagging

Joint models have been recently very popular in NLP tasks. PoS tagging is also one of those tasks that is performed jointly with other tasks, such as morphological segmentation and morphological disambiguation.

Qiu et al. [40] present a joint PoS tagging and segmentation model that integrates two Markov chains. While one chain is used for segmentation, the other is used for PoS tagging. It is tested on Chinese segmentation and the model outperforms other traditional methods.

Another joint PoS tagging and morphological segmentation model is presented by Sirts and Alumäe [44]. It is a non-parametric Bayesian model that generates the word's tag and segmentation jointly. HMM parameters are estimated by using Hierarchical Dirichlet Process (HDP). Gibbs sampling and Metropolis-Hastings sampling are used for the inference.

Can and Manandhar [11] present a Dirichlet process model for joint POS tagging and morphological segmentation. It is a generative model where the authors generate the POS tags, stems, and suffixes of words jointly. A mixture model is adopted for POS tagging by using the tags of the contextual words. A Dirichlet process model is adopted for morphology learning where stems may belong to any POS tag. Our model is similar to that model since both models generate stems based on the POS tags. However, two models are different in terms of the mathematical model they are based on. In our model, POS tagging is investigated more like a sequence labelling problem by using a Hidden Markov Model, whereas in their model it is a mixture model. Moreover, in the current model we do not adopt a Dirichlet process model for morphology, instead we generate stems and affixes based on the POS tags.

Sirts et al. [45] use distributional and morphological information in PoS tagging. The distributional and morphological information is combined in a non-parametric Bayesian model based on distance-dependent Chinese Restaurant Process (ddCRP). They extend the Chinese Restaurant Process (CRP) and define a distribution over partitions of data points. Word embeddings are used to represent the distributional characteristics of the words.

3 MODELS FOR JOINT LEARNING OF POS TAGS AND STEMS

We adopt the Bayesian HMM model of Goldwater and Griffiths [19], which is introduced for solely PoS tagging. We adopt this model as a baseline model in this article. We introduce different models that are based on their model, however they all jointly learn PoS tags and stems in an unsupervised framework.

After the terminology is given in Section 3.1, we explain our models in Section 3.2.

3.1 Terminology

A brief description of the terms used in the rest of the article is as follows:

- Segmentation of a word, $w = s + m$, where s is the stem and m is the suffix of the word w,
- t_i, w_i, s_i, m_i denote the ith tag, word, stem and suffix in the corpus respectively,
- Identity function $I(.)$ is a function that returns 1 if its argument is true and returns 0 if its argument is false,
- $n_{(t_i, w_i)}$ is the frequency of the tag-word pair (t_i, w_i),
- $n_{(t_i, s_i)}$ is the frequency of the tag-stem pair (t_i, s_i),
- $n_{(t_i, m_i)}$ is the frequency of the tag-suffix pair (t_i, m_i),
- $cos(s_i, w_i)$ is the cosine similarity between the word embeddings of s_i and w_i,
- $n_{(t_{i-2}, t_{i-1})}$ is the frequency of the tag bigram $< t_{i-2}, t_{i-1} >$,
- $n_{(m_{i-2}, m_{i-1})}$ is the frequency of the suffix bigram $< m_{i-2}, m_{i-1} >$,
- $n_{(t_{i-2}, t_{i-1}, t_i)}$ is the frequency of the tag trigram $< t_{i-2}, t_{i-1}, t_i >$,
- $n_{(m_{i-2}, m_{i-1}, m_i)}$ is the frequency of the suffix trigram $< m_{i-2}, m_{i-1}, m_i >$,
- $\mathbf{t}_{-i}$ is the current values of all tags except t_i,
- $\mathbf{w}_{-i}$ is the current values of all words except w_i,
- $\mathbf{s}_{-i}$ s the current values of all stems except s_i,
- $\mathbf{m}_{-i}$ is the current values of all suffixes except m_i,
- W_{t_i}, S_{t_i}, and M_{t_i} are the total number of word, stem, and suffix types respectively emitted from t_i,
- T is the size of the tag set,
- *Multinomial Distribution* $Mult(\omega^t)$ is the emission distribution in the form of a Multinomial distribution with parameters $\omega^{(t)}$,
- $Mult(\tau^{(t,t')})$ is the transition distribution with parameters $\tau^{(t,t')}$,
- $Mult(\psi^{(t)})$ is the suffix emission distribution in the form of a Multinomial distribution with parameters $\psi^{(t)}$,
- $\omega^{(t)}$ is generated by $Dirichlet(\beta)$ with hyperparameters β,
- $\tau^{(t,t')}$ is generated by $Dirichlet(\alpha)$ with hyperparameters α,
- $\psi^{(t)}$ is generated by $Dirichlet(\gamma)$ with hyperparameters γ.

3.2 Mathematical Model Definition

Goldwater and Griffiths [19] propose a word-based Bayesian HMM model that extends the standard HMM model by adding prior distributions to the model parameters to learn a distribution of parameters in a Bayesian setting rather than the point estimates of the parameters. We extend the baseline Bayesian HMM model to learn the PoS tags and stems jointly in a fully unsupervised setting. To this end, we propose different versions by adopting different dependencies in the data.

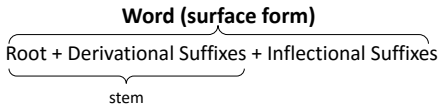

Fig. 1. A typical word structure in an agglutinative language.

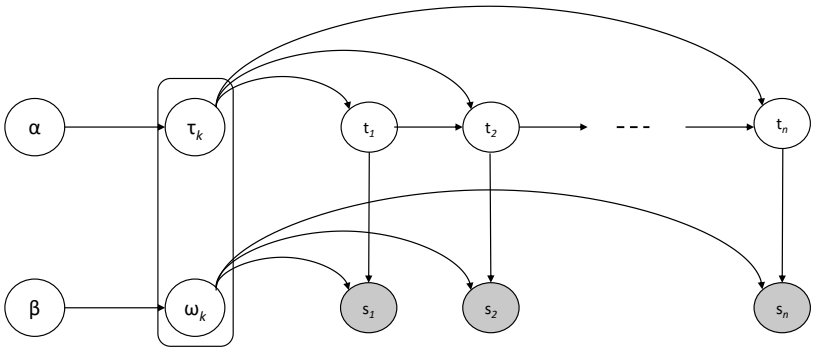

Fig. 2. The plate diagram of the stem-based Bayesian HMM.

3.2.1 Stem-based Bayesian HMM (Bayesian S-HMM). The word structure in an agglutinative language is shown in Figure 1. Stem of a word is obtained by removing the inflectional suffixes from the stem. This postulates that a word and its stem must have the same PoS tag.

The PoS tag of a word is a strong indicator for its stem. For example, if *gelecek* is a noun, then stripping off *-ecek* is a stemming error because *-ecek* is a derivational suffix here, whereas if *gelecek* is a verb, then stripping off *-ecek* is correct because *-ecek* is an inflectional suffix.

Additionally, using stems reduces the emission sparsity in agglutinative languages. Thus, we extend the word-based HMM model with the stem emissions. The mathematical model is given as follows:

$$\begin{aligned} t_i|t_{i-1}, t_{i-2} = t^{'}, \tau^{(t,t^{'})} &\propto Mult(\tau^{(t,t^{'})}) \\ s_i|t_i = t, \omega^{(t)} &\propto Mult(\omega^{(t)}) \\ \tau^{(t,t^{'})}|\alpha &\propto Dirichlet(\alpha) \\ \omega^{(t)}|\beta &\propto Dirichlet(\beta) \end{aligned} \tag{1}$$

$Mult(\omega^t)$ differs from the baseline model. It is the stem emission distribution in the form of a Multinomial distribution with parameters $\omega^{(t)}$. In other words, a stem is emitted from each HMM state. The plate diagram of the model is given in Figure 2.

Based on the mathematical model, the conditional probability of a tag and a stem are defined as follows:

$$P(t_i|\mathbf{t_{-i}}, \alpha) = \frac{n_{(t_{i-2},t_{i-1},t_i)} + \alpha}{n_{(t_{i-2},t_{i-1})} + T\alpha} \tag{2}$$

$$P(s_i|\mathbf{t_{-i}}, \mathbf{s_{-i}}, \beta) = \frac{n_{(t_i,s_i)} + \beta}{n_{(t_i)} + S_{t_i}\beta} \tag{3}$$

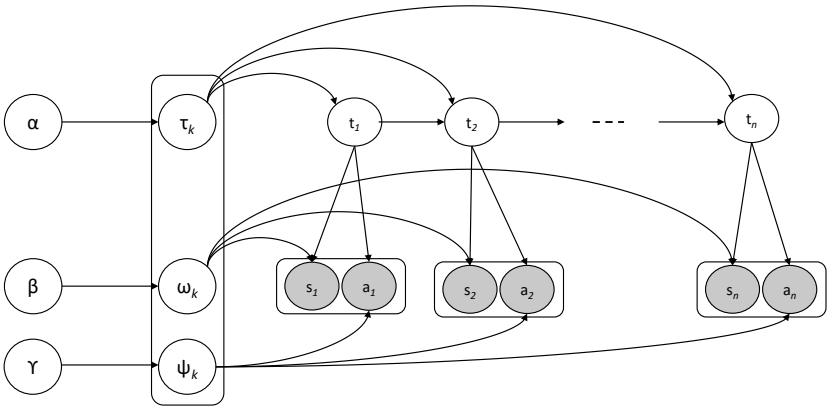

Fig. 3. The plate diagram of the stem and suffix-based Bayesian HMM.

The inference involves estimating the following posterior distribution:

$$P(\mathbf{t}, \mathbf{s}|\alpha, \beta) \quad \propto \quad P(\mathbf{s}|\mathbf{t}, \beta)P(\mathbf{t}|\alpha) \tag{4}$$

We use Gibbs sampling for the inference. All tags are randomly initialized and all words are split into two segments randomly as a stem and a suffix at the beginning of the inference. In each iteration of the algorithm, a tag and a stem are sampled for each word from the following sampling distribution:

$$\begin{aligned} P(t_i, s_i|\mathbf{t_{-i}}, \mathbf{s_{-i}}, \alpha, \beta) \quad = \quad & \frac{n_{(t_i,s_i)} + \beta}{n_{t_i} + S_{t_i}\beta} \cdot \frac{n_{(t_{i-2},t_{i-1},t_i)} + \alpha}{n_{(t_{i-2},t_{i-1})} + T\alpha} \\ & \cdot \frac{n_{(t_{i-1},t_i,t_{i+1})+I(t_{i-2}=t_{i-1}=t_i=t_{i+1})} + \alpha}{n_{(t_{i-1},t_i)} + I(t_{i-2} = t_{i-1} = t_i) + T\alpha} \\ & \cdot \frac{n_{(t_i,t_{i+1},t_{i+2})+I(t_{i-2}=t_i=t_{i+2},t_{i-1}=t_{i+1})+I(t_{i-1}=t_i=t_{i+1}=t_{i+2})} + \alpha}{n_{(t_i,t_{i+1})} + I(t_{i-2} = t_i, t_{i-1} = t_{i+1}) + I(t_{i-1} = t_i = t_{i+1}) + T\alpha} \end{aligned} \tag{5}$$

Sampling a tag affects three tag trigrams in the model at the same time because each tag occurs in three consecutive tag trigrams. Therefore, the changes are taken into account with the identity function. This process is repeated until the system converges.

3.2.2 Stem & Suffix-based Bayesian HMM (Bayesian SM-HMM). The ending of a word usually gives a clue about its syntactic category. For example, words ending with *-ly* are usually adverbs in English. We extended the stem-based Bayesian HMM model by adding suffix emissions in addition to the stem emissions in the model:

$$\begin{aligned} m_i|t_i = t, \psi^{(t)} \quad & \propto \quad Mult(\psi^{(t)}) \\ \psi^{(t)}|\gamma \quad & \propto \quad Dirichlet(\gamma) \end{aligned} \tag{6}$$

The plate diagram of the model is given in Figure 3.

Based on the mathematical model, the conditional probability of a suffix is defined as follows:

$$P(m_i|\mathbf{t_{-i}}, \mathbf{m_{-i}}, \gamma) \quad = \quad \frac{n_{(t_i,m_i)} + \gamma}{n_{(t_i)} + M_{t_i}\gamma} \tag{7}$$

Table 1. Cosine similarity of derived and inflected words from *araba* and word *araba*

Word	Cosine similarity
arabanızı	0.514376077418
arabayı	0.672099882553
arabalar	0.630295555647
arabalara	0.614237023072
arabaları	0.616125127335
arabanın	0.616908095499
arabada	0.578588391368
arabacı	0.256999428043
arabayı	0.672099882553
arabacılar	0.258639561905
arabacılık	0.231535188189

The conditional probability for the stem and the tag are the same as given in the stem-based model.

The inference involves estimating the following posterior distribution:

$$P(\mathbf{t}, \mathbf{s}, \mathbf{m}|\alpha, \beta, \gamma) \quad \propto \quad P(\mathbf{s}|\mathbf{t}, \beta)P(\mathbf{m}|\mathbf{t}, \gamma)P(\mathbf{t}|\alpha) \tag{8}$$

Here, we assume that stems and suffixes are independent from each other. Again we use Gibbs sampling for the inference. Word tags are randomly initialized and all words are split into two segments uniformly at the beginning of the inference. In each iteration of the algorithm, a tag, a stem and a suffix are sampled for each word from the following sampling distribution:

$$\begin{aligned} P(t_i, s_i, m_i|\mathbf{t_{-i}}, \mathbf{s_{-i}}, \mathbf{m_{-i}}, \alpha, \beta, \gamma) \quad = \quad & \frac{n_{(t_i,s_i)}\beta}{n_{t_i} + S_{t_i}\beta} \cdot \frac{n_{(t_{i-2},t_{i-1},t_i)} + \alpha}{n_{(t_{i-2},t_{i-1})} + T\alpha} \\ \cdot \quad & \frac{n_{(t_{i-1},t_i,t_{i+1})+I(t_{i-2}=t_{i-1}=t_i=t_{i+1})} + \alpha}{n_{(t_{i-1},t_i)} + I(t_{i-2} = t_{i-1} = t_i) + T\alpha} \\ \cdot \quad & \frac{n_{(t_i,t_{i+1},t_{i+2})+I(t_{i-2}=t_i=t_{i+2},t_{i-1}=t_{i+1})+I(t_{i-1}=t_i=t_{i+1}=t_{i+2})} + \alpha}{n_{(t_i,t_{i+1})} + I(t_{i-2} = t_i, t_{i-1} = t_{i+1}) + I(t_{i-1} = t_i = t_{i+1}) + T\alpha} \\ \cdot \quad & \frac{n_{(t_i,m_i)} + \gamma}{n_{t_i} + M_{t_i}\gamma} \end{aligned} \tag{9}$$

3.2.3 Stem-based Bayesian HMM using Neural Word Embeddings (Bayesian CS-HMM). Unlike derivation, inflectional suffixes do not make a significant change in the meaning of a word. Inflectional suffixes only fulfil some syntactic functions, such as gender and tense in verbs. We add some semantic information to enhance the stem-based Bayesian model. To this end, we include neural word embeddings obtained from word2vec [33][1] to estimate the similarity between the stem and the word form. The more similar the stem embedding and the word embedding are, more likely to have an inflection in the word form, rather than a derivation.

Cosine similarity between the neural word embeddings of *araba* (*the car*) and some other word forms that are inflected or derived from *araba* are given in Table 1. It is clearly seen that the words that share the same stem have a higher cosine similarity with *araba* unlike the derived words (i.e. *arabacı*, *arabacılar*, arabacılık).

[1]The details of the datasets used for training the word2vec [33] is given in Appendix A.

We use the cosine similarity between the word embeddings of the stem and the word itself as prior information in the model. The mathematical model is the same as the stem-based Bayesian HMM model given in Section 3.2.1. The sampling distribution of t_i and s_i under this model is the same as Equation 5.

We use the cosine similarity as a factor to increase/decrease the probability of a {stem, tag} pair proportionally to the cosine similarity:

$$\sum_{t_i, s_i} P(t_i, s_i | \mathbf{t_{-i}}, \mathbf{s_{-i}}, \alpha, \beta) cos(s_i, w_i) = 1 \tag{10}$$

3.2.4 Stem & Suffix-based Bayesian HMM using Neural Word Embeddings (Bayesian CSM-HMM). In this model, we extend the stem and suffix-based Bayesian HMM model by again adding semantic information obtained from the neural word embeddings analogously to the previous model. Therefore, the mathematical model is the same as the stem-suffix-based Bayesian HMM model given in Section 3.2.2. The new conditional distribution of t_i, s_i and m_i becomes the same with 9.

We again use the cosine similarity as a factor to increase/decrease the probability of a {stem, suffix, tag} tuple proportionally to the cosine similarity:

$$\sum_{t_i, s_i} P(t_i, s_i, m_i | \mathbf{t_{-i}}, \mathbf{s_{-i}}, \mathbf{m_{-i}}, \alpha, \beta, \gamma) cos(s_i, w_i) = 1 \tag{11}$$

4 EXPERIMENTS AND RESULTS

In this section, we present the experimental results obtained from the joint PoS tagging and stemming models described in the previous section[2]. We compared the experimental results obtained for PoS tagging with Anchor HMM[3] [46], Brown Clustering[4] [9], and the word-based Bayesian HMM model [19]. We compared the results obtained for stemming with HPS[5] [10], Linguistica[6] [18], and Morfessor FlatCat[7] [21]. We used the same datasets given below to train these models.

We describe the datasets used in the experiments in Section 4.1 and the parameter settings of the models are described in Section 4.2. Details of the evaluation metrics for PoS tagging and stemming are given in Section 4.3 and the final results are presented in Section 4.4.

4.1 Datasets

We did experiments on five languages: Turkish, Hungarian, Finnish, Basque, and English. The training sets used for these languages are as follows:

- **Turkish :** METU Treebank [35] is a Turkish treebank that is collected from newspapers, journal issues, and books. The treebank involves 5,620 sentences and 53,798 tokens.
- **Finnish :** FinnTreeBank [50] is a manually annotated Finnish dataset that involves around 19,000 sentences and sentence fragments, and 162,000 word forms. We used the first 24K words from the Finnish dataset.

[2]The source code of all of the models described in this article can be found at: https://github.com/necvabolucu/Joint-PoS-tagging-and-Stemming
[3]Anchor HMM: https://github.com/karlstratos/anchor
[4]Brown Clustering: http://www.cs.berkeley.edu/~pliang/software/brown-cluster-1.2.zip
[5]HPS: http://liks.fav.zcu.cz/HPS/
[6]Linguistica: http://linguistica.uchicago.edu/
[7]Morfessor FlatCat: https://github.com/aalto-speech/flatcat

Table 2. Datasets used in the experiments

Language	Dataset	Tagset size
Basque	UD Dependency Treebank [34]	16
English	Penn Treebank [28]	45
English	UD Dependency Treebank [34]	17
Finnish	FinnTreeBank [50]	14
Hungarian	UD Dependency Treebank [34]	16
Turkish	METU Treebank [35]	31

- **Hungarian, Basque and English :** UD Dependency Treebank [34] is a multilingual treebank. We used the Hungarian, Basque, and English portions of the treebank. We used the first 24K words for each language in the experiments.
- **English :** Penn Treebank [28] is an English treebank that is collected from the Air Traffic Information System, the Wall Street Journal (WSJ), the Brown Corpus, Switchboard, and a variety of other sources. We used the first 24K words from the treebank for the experiments.

We reduced the PoS tagset size to 12 based on the universal PoS tagset[8] [38] to be able to compare the accuracy of PoS tagging across different languages. The datasets and the actual tagset size of all languages are given in Table 2.

4.2 Parameters

We manually set the hyperparameters in the experiments. We assigned the values of the hyperparameters based on a series of experiments. In order to alleviate the affect of the hyperparameter values, we did the experiments using six hyperparameter sets for all languages. The hyperparameter sets used in the experiments are given below:

(1) set α=0.003 β=1 γ=0.003
(2) set α=0.003 β=0.1 γ=0.003
(3) set α=0.001 β=1 γ=0.001
(4) set α=0.001 β=0.1 γ=0.001
(5) set α=0.03 β=1 γ=0.03
(6) set α=0.03 β=0.1 γ=0.03

We ran each experiment for 5000 iterations in Gibbs sampling.

4.3 Evaluation Metrics

We did the evaluation for PoS tagging and stemming separately by using different evaluation metrics:

4.3.1 PoS tagging. We used four evaluation metrics to evaluate the PoS tagging results: many-to-one, one-to-one, normalized mutual information (NMI), and variation of information (VI). Many-to-one [23] maps each result tag to the gold standard tag that has the maximum number of common words with the result tag. One gold standard tag can be assigned to more than one result tag. One-to-one [22] is computed similarly, however each gold standard tag is restricted to having only one result tag.

[8]Reduced tagsets based on [38] are given in Appendix B.

NMI [47] normalizes the symmetric measure of two clusterings C_g (the gold clustering) and C_r (the result clustering). It is defined in terms of entropy H and mutual information I:

$$NMI(C_r, C_g) = \frac{I(C_r, C_g)}{\sqrt{H(C_r)H(C_g)}} \tag{12}$$

VI [30] is a distance metric between two clusterings C_g and C_r:

$$VI(C_r, C_g) = H(C_r|C_g) + H(C_g|C_r) \tag{13}$$

4.3.2 Stemming. We used accuracy and Frakes and Fox Similarity Metric (FSM) [14] to evaluate the stemming results. Accuracy measures the correctness of stems:

$$A = \frac{C_s}{C_w} \tag{14}$$

where C_s is the number of correct stems and C_w is the total number of words in the corpus.

FSM is used to evaluate the strength of the stemming:

$$N_a = \frac{N_w}{N_s} \tag{15}$$

N_a is the average number of words per the conflation class, N_w refers to the number of unique words before stemming, and N_s is the number of unique stems after stemming.

4.4 Results

We present PoS tagging and stemming results separately for five languages. We did experiments for all parameter settings but we present only the highest scores for the sake of easiness and readability. In order to show the affect of the parameters in the model, the many-to-one scores obtained from different parameter settings for PoS tagging in Turkish is given in Table 3. We assign α (transition hyperparameter) and γ (suffix emission hyperparameter) low values because the variety in the transitions and the suffixes is comparably less than the variety in the stems. Therefore, we assign β higher values than α and γ. When we emit stems and suffixes at the same time (in Bayesian SM-HMM, Bayesian CSM-HMM), the highest scores are obtained when $\beta = 1$. In other words, when a large number of stem types are permitted to be emitted from the HMM, a small number of suffix types are emitted from the HMM, which linguistically makes more sense. When suffix emissions are not used, the highest scores are obtained when $\beta = 0.1$.

We did also a similar study to show the affect of the hyperparameters in stemming. The accuracy obtained from different parameter settings for stemming in Turkish is given in Table 4. Stemming results show that models with semantic information (Bayesian CS-HMM and Bayesian CSM-HMM) give the best results with relatively low β=0.1 and high $\alpha = \gamma$=0.03 because semantic information restricts to have more variety in stem types that can be emitted. The other two proposed models (Bayesian S-HMM and Bayesian SM-HMM) give the best results under relatively high value of β=1 and low value of $\alpha = \gamma$=0.003.

PoS tagging results are given in Table 5 for all languages. The highest scores among 6 parameter settings are given in the table. The highest scores in PoS tagging were obtained from the 6th parameter setting for Turkish, Basque, and English. However, the highest scores were obtained from the 5th parameter setting for Hungarian and Finnish.

For Turkish language, the highest many-to-one accuracy is obtained from the stem-based Bayesian S-HMM, whereas the highest one-to-one accuracy, NMI and VI measures are obtained from the stem-based Bayesian CSM-HMM that uses neural word embeddings. This shows that using semantic similarity helps in detecting the correct PoS tag of a given

Table 3. Many-to-one scores obtained from different hyperparameter values for Turkish PoS tagging

		Value of α=γ		
	Value of β	0.003	0.001	0.03
Bayesian HMM	1	54.92	55.81	55.51
	0.1	55.44	55.31	**56.79**
Bayesian S-HMM	1	53.36	51.58	53.89
	0.1	52.04	53.64	**64.43**
Bayesian SM-HMM	1	**58.07**	56.05	56.97
	0.1	56.46	56.70	56.43
Bayesian CS-HMM	1	52.27	51.81	54.95
	0.1	50.97	53.15	**57.38**
Bayesian CSM-HMM	1	60.09	**60.53**	60.10
	0.1	59.36	59.79	59.32

Highest scores obtained from different parameter settings for PoS tagging in Turkish are given in bold in the table.

Table 4. Accuracy scores obtained from different hyperparameter values for Turkish stemming

		Value of α=γ		
	Value of β	0.003	0.001	0.03
Bayesian S-HMM	1	**47.55**	47.30	47.45
	0.1	47.51	47.46	47.29
Bayesian SM-HMM	1	**34.97**	34.97	34.97
	0.1	34.97	34.97	34.96
Bayesian CS-HMM	1	57.31	57.47	57.85
	0.1	57.76	63.71	**63.83**
Bayesian CSM-HMM	1	39.51	39.50	39.62
	0.1	40.55	40.22	**41.08**

The highest scores obtained from parameter settings for stemming in Turkish are written bold in the table.

word. In any case, our Turkish results are higher than the results of the word-based Bayesian HMM model, Brown Clustering, and Anchor HMM.

The stem and suffix-based Bayesian SM-HMM gives the highest scores for all evaluation metrics for the Hungarian language. This shows that the suffix of a word is a strong indicator for the PoS tag of a word.

The highest many-to-one accuracy and NMI are obtained from the stem and suffix-based Bayesian SM-HMM in Finnish. Brown Clustering gives the highest scores for one-to-one accuracy and VI in Finnish.

Brown Clustering gives the highest scores in many-to-one accuracy, NMI, and VI in Basque, whereas stem-based Bayesian CS-HMM that uses semantic similarity gives the highest score for one-to-one. However, our overall results in Basque are very close to the scores obtained from Brown Clustering.

In English, Brown Clustering gives the highest scores for all evaluation metrics. We assume a concatenative morphology in our model. The comparably lower scores in English can be because of the irregular words in the language.

It can be clearly seen from the results that using stems rather than words improves the PoS tagging results for agglutinative languages. This also clarifies the lower results obtained from English. When the overall results for all languages are evaluated, it can be seen that stem-based Bayesian SM-HMM and stem-based Bayesian CSM-HMM that uses neural word embeddings are significantly better than the other proposed models.

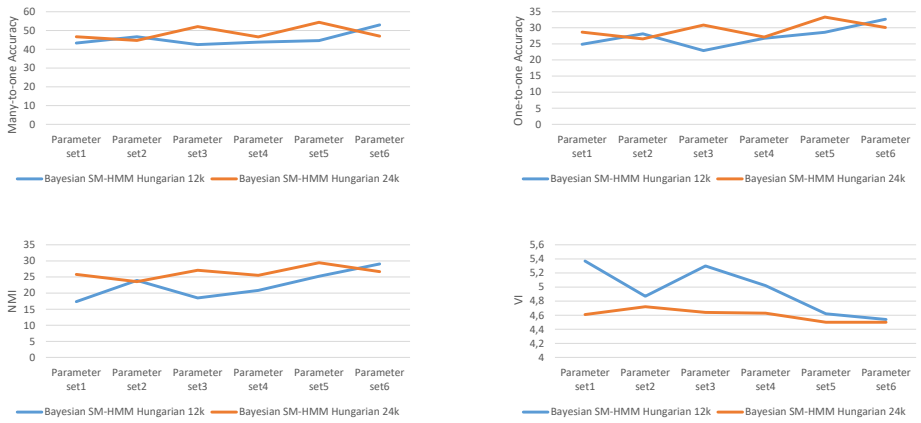

Fig. 4. Impact of the dataset size for PoS tagging in Hungarian for the Bayesian SM-HMM model.

Using neural word embeddings improves the results in Turkish, whereas there is not a significant improvement in Hungarian and Finnish results when the semantic similarity is used. This can be also due to the size of the training set in Hungarian and Finnish used for word2vec [33].

The accuracy obtained from stemming for five languages is given in Table 6. We compare our stemming results with HPS [10], Linguistica [18] and Morfessor [21]. Stem-based Bayesian CS-HMM using semantic similarity performs better than the other models for all five languages. This shows that using semantic similarity also improves the stemming results. We have the highest scores in Turkish when compared to other models. Linguistica gives the highest scores for Hungarian, Finnish and English, whereas Morfessor gives the highest accuracy for Basque.

Again the size of the training set may be also the reason for having comparably lower scores in stem-based Bayesian CS-HMM using semantic similarity for Hungarian, Finnish, Basque, and English. The training set used for word2vec [33] in Turkish is very large compared to other languages.

Stemming strength obtained from all models are given in Table 7 for five languages. When we compare Table 6 and 7, it can be seen that there is a correlation between the accuracy and the strength.

In order to show the impact of the dataset size in Hungarian, the results for 12K and 24K datasets are given for PoS tagging and stemming in Figure 4 and Figure 5 respectively. PoS tagging results of the 24K dataset are higher than the results obtained from the 12K dataset. However, stemming results show that the dataset size does not make a significant improvement on the stemming performance. Moreover, stemming results obtained from the smaller dataset are higher than the results obtained from the larger dataset.

This is to show that the performance of our model does not change significantly according to the different dataset sizes in Hungarian. Similar results are also obtained for other languages. The results for 12K and 24K datasets for PoS tagging and stemming in Finnish, Basque and English are given in Appendix C in Figure 6 and Figure 7. Results obtained for the other datasets are similar to that of Hungarian. When we analyze the stemming results, it is clearly seen that stem-based models do not require a large dataset, while models using stems and affixes require a larger dataset.

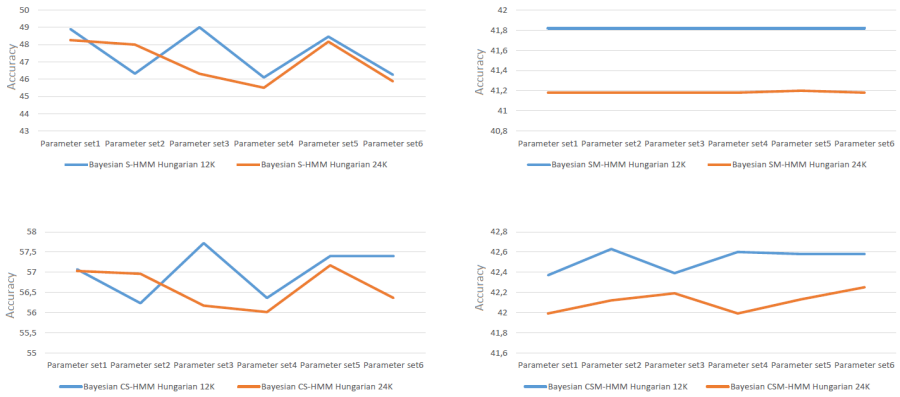

Fig. 5. Impact of the dataset size for stemming in Hungarian for all models.

5 DISCUSSION

Out-of-vocabulary issue is one of the challenges in many natural language processing tasks. In this paper, we introduced a joint learning framework for PoS tagging and stemming that resolves the out-of-vocabulary issue in PoS tagging, while inferring stems simultaneously. We experimented with different versions where we utilize only the stem information, stem and affix information, or the semantic relatedness between the affixed form and the stem based on the idea that meaning of a word is protected even though an inflectional suffix is attached.

We observed that the highest PoS tagging results were obtained from Turkish that is morphologically the richest among all other languages. It shows that incorporating the stem information with the syntactic task improves the inference of the PoS tags for a morphologically rich language, which was one of the main goals of this work. The situation is a bit different in morphologically poor languages such as English. The joint models do not perform very well on English especially for the PoS tagging task. However, the best stemming results were achieved for the English language. It may be due to the poorly inflected structure of English, which enables learning the stems more accurately compared to other morphologically rich languages such as Turkish and Hungarian. We can conclude that the joint learning of stems and PoS tags particularly perform well on agglutinative languages for the PoS tagging task, whereas it performs well on morphologically poor languages for the stemming task. One limitation of the model is the irregular word forms since we assume a concatenative morphology in this work.

We introduced several different joint models that adopt different dependencies in the corpus, such as the dependency between the PoS tag and the suffix, or the dependency between the stem and the PoS tag. The overall experimental results show that stem and suffix-based Bayesian HMM model using neural word embeddings outperforms other models for the POS tagging task, whereas stem-based Bayesian HMM model using neural word embeddings outperforms other models for the stemming task.

Additionally, we use semantic similarity between the stems and words to discover the inflectional morphology since the inflectional suffixes do not change the meaning of a word. To this end, we use the neural word embeddings obtained from word2vec [33]. The results show that using semantic information makes a significant improvement in both stemming and PoS tagging.

We compare our models with other PoS tagging and stemming models. Our joint models perform better PoS tagging than the word-based Bayesian HMM model [19], Brown Clustering [9], and Anchor HMM model [46] for three agglutinative languages: Turkish, Finnish, Hungarian. The proposed models also perform better than HPS [10] and Morfessor FlatCat [21] in stemming.

6 CONCLUSION AND FUTURE WORK

In this article, we present unsupervised joint PoS tagging and stemming models for agglutinative languages. To our knowledge, this is the first attempt to combine PoS tagging and stemming tasks jointly in a fully unsupervised framework. We did experiments on five languages: Turkish, Hungarian, Finnish, Basque, and English. Although we aim for agglutinative languages, the results obtained from non-agglutinative languages show that the models can be applied to any other language as well. The performance of our models on five languages shows that using stems instead of words outperforms word-based PoS tagging models for agglutinative languages.

In this work, we assume a concatenative morphology. We leave the irregular words as a future goal. In the future, we will aim to learn the transformations between words in order to handle irregular words. We also aim to exploit our results in a high order NLP task such as text categorization for an extrinsic evaluation.

ACKNOWLEDGEMENTS

This research is supported by the Scientific and Technological Research Council of Turkey (TUBITAK) with the project number EEEAG-115E464.

APPENDIX

A WORD2VEC TRAINING SETS

The details about the datasets used to learn the distributed representations of words by word2vec [33] for five languages are as follows:

English : The corpus with the first one billion characters from Wikipedia. (http://mattmahoney.net/dc/text8.zip)

Finn Treebank [50]: The treebank involves around 19,000 sentences and sentence fragments, and 162,000 word forms.

The Basque UD Treebank [34]: The dataset is a part of the Basque Dependency Treebank (BDT) [2]. The treebank consists of 8,993 sentences and 121,443 tokens.

The Hungarian UD Treebank [34]: The treebank is derived from the Szeged Dependency Treebank [49]. It contains 1,299 sentences and 42,032 words.

Turkish Boun Corpus [41]: It is a manually collected web corpus. It involves 423M words and 491M tokens.

B UNIVERSAL TAGSET MAPPING

The mapping of the original tagset to the Universal PoS tagset [38] is given for the Penn Treebank and FinnTreeBank in Table 8, for the UD Treebank for Basque, Hungarian and English in Table 9, and for the Metu-Sabanc Turkish Treebank in Table 10.

C EXPERIMENTS FOR THE DATASET SIZE

Here, we present PoS tagging and stemming results for Finnish, Basque, and English in order to illustrate the performance of the models according to the dataset size (12K and 14K). We have used VI for PoS tagging and Accuracy for stemming in the figures. The results are given in Figure 6-7.

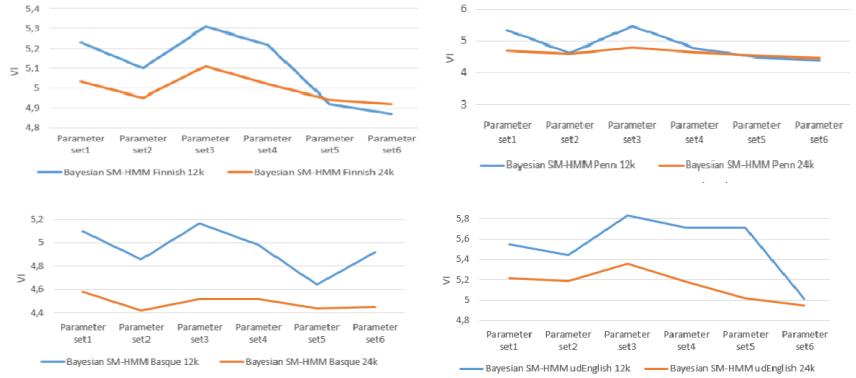

Fig. 6. Impact of the dataset size for PoS tagging in Finnish, Basque and English for the Bayesian SM-HMM model.

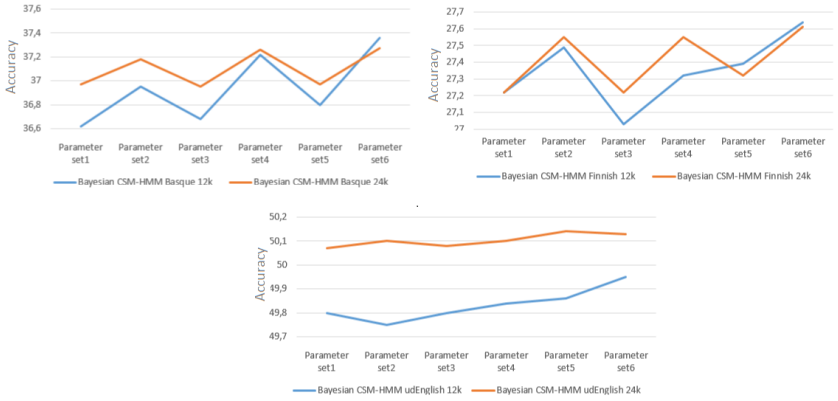

Fig. 7. Impact of the dataset size for stemming in Finnish, Basque and English for all models.

Table 5. PoS tagging results for all languages

	Many-to-one	One-to-one	NMI	VI
(a) Turkish				
Bayesian S-HMM	**64.43**	26.33	29.22	5.54
Bayesian SM-HMM	56.43	28.65	29.39	4.40
Bayesian CS-HMM	57.38	28.87	24.64	4.72
Bayesian CSM-HMM	59.32	**30.93**	**30.95**	**4.30**
Bayesian HMM [19]	56.79	27.31	23.25	4.79
Brown Clustering [9]	54.91	30.70	26.78	4.47
Anchor HMM [46]	58.82	-	-	-
(b) Hungarian				
Bayesian S-HMM	39.38	18.74	11.25	5.76
Bayesian SM-HMM	**54.38**	**33.33**	**29.46**	**4.50**
Bayesian CS-HMM	44.12	23.79	14.56	5.51
Bayesian CSM-HMM	51.20	29.95	26.17	4.71
Bayesian HMM [19]	43.92	22.41	14.75	5.48
Brown Clustering [9]	50.89	33.25	28.65	4.57
Anchor HMM [46]	48.86	-	-	-
(c) Finnish				
Bayesian S-HMM	42.70	21.78	11.64	5.49
Bayesian SM-HMM	**48.46**	23.45	**20.24**	4.94
Bayesian CS-HMM	42.61	22.42	12.62	5.43
Bayesian CSM-HMM	47.78	22.79	20.19	4.93
Bayesian HMM [19]	43.21	23.39	13.14	5.39
Brown Clustering [9]	47.95	**30.13**	17.62	**4.92**
Anchor HMM [46]	43.73	-	-	-
(d) Basque				
Bayesian S-HMM	57.38	29.95	23.63	4.66
Bayesian SM-HMM	57.31	28.49	26.64	4.45
Bayesian CS-HMM	58.13	**30.41**	23.90	4.66
Bayesian CSM-HMM	58.37	29.48	27.65	4.36
Bayesian HMM [19]	57.07	29.54	22.47	4.75
Brown Clustering [9]	**60.63**	30.37	**28.71**	**4.31**
Anchor HMM [46]	58.20	-	-	-
(e) English				
Bayesian S-HMM	30.43	23.34	15.00	6.01
Bayesian SM-HMM	39.69	32.55	28.17	4.95
Bayesian CS-HMM	37.71	31.30	22.99	5.40
Bayesian CSM-HMM	39.06	32.95	28.55	4.95
Bayesian HMM [19]	36.19	27.89	19.96	5.65
Brown Clustering [9]	**51.97**	**47.32**	**40.25**	**4.14**
Anchor HMM [46]	48.79	-	-	-

Results written bold in the table are the highest scores among 6 parameter settings are given in the table for PoS tagging of all languages.

Table 6. Stemming results for all languages

Models	Turkish	Hungarian	Finnish	Basque	English
Bayesian S-HMM	47.29	48.17	28.28	31.92	49.30
Bayesian SM-HMM	34.96	41.20	26.40	33.54	47.82
Bayesian CS-HMM	**63.83**	57.17	38.94	48.72	78.99
Bayesian CSM-HMM	41.08	42.13	27.13	37.27	50.13
HPS [10]	53.79	58.98	27.18	50.06	75.21
Linguistica [18]	52.33	**70.12**	**45.40**	53.17	**83.84**
Morfessor [21]	52.06	45.89	25.93	**55.50**	63.05

Highest accuracy results obtained with 6th parameter setting are written bold in the table for stemming of all languages.

Table 7. Stemming strength for all languages

Models	Turkish	Hungarian	Finnish	Basque	English
Bayesian S-HMM	0.75	0.65	0.47	0.47	0.74
Bayesian SM-HMM	0.60	1.14	0.68	0.71	1.58
Bayesian CS-HMM	**0.89**	0.76	0.53	0.57	**1.84**
Bayesian CSM-HMM	0.69	1.12	**0.68**	0.71	1.25
HPS [10]	0.81	1.00	0.58	**0.77**	0.79
Linguistica [18]	0.76	**1.21**	0.62	0.68	1.17
Morfessor [21]	0.77	0.41	0.54	0.66	1.02

Highest results for stemming strength of all languages are written in bold.

Table 8. The mapping of the Universal tagset to the Penn and Finn Treebank tagset

Universal tagset	Penn Treebank tagset	FinnTreeBank tagset
VERB	VBP,VBD,VBG,VBN,VB,VBZ,MD	V
PRON	WP,PRP, *PRP*, *WP*	Pron
PUNCT	(“),(,),-LRB-,-NONE-,-RRB-,(.),(:),(”),$	Punct
PRT	RP,TO	Pcle
DET	WDT,EX,PDT,DT	Det
NOUN	NN,NNP,NNPS,NNS	N
ADV	RB,RBR,WRB,RBS	Adv
ADJ	JJ,JJS	A
UNKNOWN	FW,UH	Symb, Foreign, Interj
ADP	IN	Adp
NUM	CD	Num
CONJ	CC	C

Table 9. The mapping of the Universal tagset to the UD Basque, Hungarian and English Treebank tagset

Universal tagset	Basque	Hungarian	English
VERB	VERB, AUX	VERB, AUX	VERB, AUX
PRON	PRON	PRON	PRON
PUNCT	PUNCT	PUNCT	PUNCT
PRT	PART	PART	PART
DET	DET	DET	DET
NOUN	NOUN, PROPN	NOUN, PROPN	NOUN, PROPN
ADV	ADV	ADV	ADV
ADJ	ADJ	ADJ	ADJ
UNKNOWN	SYM, INTJ, X	X, INTJ	X, INTJ, SYM
ADP	ADP	ADP	ADP
NUM	NUM	NUM	NUM
CONJ	CONJ	CONJ , SCONJ	CONJ , SCONJ

Table 10. The mapping of the Universal tagset to the Metu-Sabancı Turkish Treebank tagset

Universal tagset	Metu-Sabancı Turkish Treebank tagset
Noun	Noun_Pron,Noun_Ins,Noun_Nom,Noun_Verb,Noun_Loc, Noun_Acc,Noun_Abl,Noun_Gen, Noun_Dat,Noun_Adj, Noun_Num,Noun_Pnon,Noun_Postp,Noun_Equ
Adj	Adj_Noun,Adj_Verb,Adj,Adj_Pron,Adj_Postp,Adj_Num
Adv	Adv_Verb,Adv_Adj,Adv_Noun,Adv
Conj	Conj
Det	Det
Interj	Interj
Ques	Ques
Verb	Verb,Negp,Verb_Noun,Verb_Postp,Verb_Adj,Verb_Adv,Verb_Verb
Postp	Postp
Num	Num
Pron	Pron,Pron_Noun
Punc	Punc

REFERENCES

[1] Giorgos Adam, Konstantinos Asimakis, Christos Bouras, and Vassilis Poulopoulos. 2010. An efficient mechanism for stemming and tagging: the case of Greek language. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 389–397.

[2] Arriola J.M. Atutxa A. Daz de Ilarraza A. Garmendia A. Oronoz M. Aduriz I.*, Aranzabe M.J. 2003. Construction of a Basque dependency treebank. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*. 201–204.

[3] Sanjeev Arora, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. 2013. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the International Conference on Machine Learning*. 280–288.

[4] Michela Bacchin, Nicola Ferro, and Massimo Melucci. 2002. The effiectiveness of a graph-based algorithm for stemming. In *Proceedings of the 5th International Conference on Asian Digital Libraries: Digital Libraries: People, Knowledge, and Technology*. Springer, 117–128.

[5] Michela Bacchin, Nicola Ferro, and Massimo Melucci. 2005. A probabilistic model for stemmer generation. *Information processing & management* 41, 1 (2005), 121–137.

[6] Michele Banko and Robert C Moore. 2004. Part of speech tagging in context. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 556.

[7] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics* 41, 1 (1970), 164–171.

[8] Chris Biemann. 2006. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics: student research workshop*. Association for Computational Linguistics, 7–12.

[9] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18, 4 (1992), 467–479.

[10] Tomáš Brychcín and Miloslav Konopík. 2015. HPS: High precision stemmer. *Information Processing & Management* 51, 1 (2015), 68–91.

[11] Burcu Can and Suresh Manandhar. 2013. Dirichlet processes for joint learning of morphology and PoS tags. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. 1087–1091.

[12] Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised POS induction: How far have we come?. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 575–584.

[13] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. A practical part-of-speech tagger. In *Proceedings of the third conference on Applied natural language processing*. Association for Computational Linguistics, 133–140.

[14] William B Frakes and Christopher J Fox. 2003. Strength and similarity of affix removal stemming algorithms. In *ACM SIGIR Forum*, Vol. 37. ACM, 26–30.

[15] Jianfeng Gao and Mark Johnson. 2008. A comparison of Bayesian estimators for unsupervised Hidden Markov Model POS taggers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 344–352.

[16] Stuart Geman and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), 721–741.

[17] John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational linguistics* 27, 2 (2001), 153–198.

[18] John Goldsmith. 2006. An algorithm for the unsupervised learning of morphology. *Natural Language Engineering* 12, 04 (2006), 353–371.

[19] Sharon Goldwater and Tom Griffiths. 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Annual meeting-association for computational linguistics*, Vol. 45. Citeseer, 744.

[20] A Goweder, H Alhami, Tarik Rashed, and A Al-Musrati. 2008. A hybrid method for stemming Arabic text. *Journal of computer Science, URL: http://eref. uqu. edu. sa/files/eref2/folder6/f181. pdf* (2008).

[21] Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 1177–1185.

[22] Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics, 320–327.

[23] Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers?. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. 296–305.

[24] Jon M Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S Tomkins. 1999. The web as a graph: measurements, models, and methods. In *International Computing and Combinatorics Conference*. Springer, 1–17.

[25] Robert Krovetz. 1993. Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 191–202.

[26] Julie Beth Lovins. 1968. Development of a stemming algorithm. *Mech. Translat. & Comp. Linguistics* 11, 1-2 (1968), 22–31.

[27] Prasenjit Majumder, Mandar Mitra, Swapan K Parui, Gobinda Kole, Pabitra Mitra, and Kalyankumar Datta. 2007. YASS: Yet another suffix stripper. *ACM transactions on information systems (TOIS)* 25, 4 (2007), 18.

[28] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics* 19, 2 (1993), 313–330.

[29] Paul McNamee and James Mayfield. 2004. Character n-gram tokenization for European language text retrieval. *Information retrieval* 7, 1-2 (2004), 73–97.

[30] Marina Meilă. 2007. Comparing clusteringsfi!?an information based distance. *Journal of multivariate analysis* 98, 5 (2007), 873–895.

[31] Massimo Melucci and Nicola Orio. 2003. A novel method for stemmer generation based on hidden Markov models. In *Proceedings of the twelfth international conference on Information and knowledge management*. ACM, 131–138.

[32] Bernard Merialdo. 1994. Tagging English text with a probabilistic model. *Computational linguistics* 20, 2 (1994), 155–171.

[33] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[34] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, and others. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. 1659–1666.

[35] Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish treebank. *Treebanks* (2003), 261–277.

[36] Jiaul H Paik, Mandar Mitra, Swapan K Parui, and Kalervo Järvelin. 2011. GRAS: An effective and efficient stemming algorithm for information retrieval. *ACM Transactions on Information Systems (TOIS)* 29, 4 (2011), 19.

[37] Fuchun Peng, Nawaaz Ahmed, Xin Li, and Yumao Lu. 2007. Context sensitive stemming for web search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 639–646.

[38] Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086* (2011).

[39] Martin F Porter. 1980. An algorithm for suffix stripping. *Program* 14, 3 (1980), 130–137.

[40] Xipeng Qiu, F Eng Ji, Jiayi Zhao, and Xuanjing Huang. 2012. Joint segmentation and tagging with coupled sequences labeling. (2012).

[41] Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2008. Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus. In *Advances in Natural Language Processing: 6th International Conference, GoTAL 2008 Gothenburg, Sweden, August 25-27, 2008 Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, 417–427.

[42] Hinrich Schütze. 1993. Part-of-speech induction from scratch. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 251–258.

[43] Manish Shrivastava, Nitin Agrawal, Bibhuti Mohapatra, Smriti Singh, and Pushpak Bhattacharya. 2005. Morphology based natural language processing tools for indian languages. In *Proceedings of the 4th Annual Inter Research Institute Student Seminar in Computer Science, IIT, Kanpur, India, April*. Citeseer.

[44] Kairit Sirts and Tanel Alumäe. 2012. A hierarchical Dirichlet process model for joint part-of-speech and morphology induction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 407–416.

[45] Kairit Sirts, Jacob Eisenstein, Micha Elsner, and Sharon Goldwater. 2014. POS induction with distributional and morphological information using a distance-dependent Chinese restaurant process. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2. 265–271.

[46] Karl Stratos, Michael Collins, and Daniel Hsu. 2016. Unsupervised part-of-speech tagging with anchor hidden markov models. *Transactions of the Association for Computational Linguistics* 4 (2016), 245–257.

[47] Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* 3, Dec (2002), 583–617.

[48] Jurgen Van Gael, Andreas Vlachos, and Zoubin Ghahramani. 2009. The infinite HMM for unsupervised PoS tagging. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*. Association for Computational Linguistics, 678–687.

[49] Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010. Hungarian Dependency Treebank.. In *LREC*, Vol. 10. Citeseer, 1855–1862.

[50] Atro Voutilainen, Tanja Purtonen, and Kristiina Muhonen. 2012. Outsourcing Parsebanking: The FinnTreeBank Project. In *Shall We Play the Festschrift Game?* Springer, 117–131.

[51] Stephen G Walker. 2007. Sampling the Dirichlet mixture model with slices. *Communications in Statistics-Simulation and Computation®* 36, 1 (2007), 45–54.

[52] Jinxi Xu and W Bruce Croft. 1998. Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems (TOIS)* 16, 1 (1998), 61–81.